(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 818 736 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.01.1998 Bulletin 1998/03

(51) Int. Cl.[6]: G06F 15/16, G06F 9/46

(21) Application number: 97111466.5

(22) Date of filing: 07.07.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV RO SI

(30) Priority: 12.07.1996 US 678891

(71) Applicant:
DIGITAL EQUIPMENT CORPORATION
Maynard, Massachusetts 01754 (US)

(72) Inventor: Huang, Arthur
Ta-Chi Taoyuan (TW)

(74) Representative: Betten & Resch
Reichenbachstrasse 19
80469 München (DE)

(54) System for assigning boot strap processor in symmetric multiprocessor computer with watchdog reset

(57) A boot strap assignment system is disclosed for a symmetric multiprocessor computer in which the role of the boot strap processor is assigned to one of the working processors by a central agent as part of power-on configuration. The system includes a system management processor which monitors the operation of the multiprocessor computer and controls a switching circuit that selectively transmits the boot strap assignment signal from the central agent to the working processors. Since the management processor monitors environmental conditions and shut-down events, it can predict the failure of working processors and assign the bootstrapping function appropriately. A watch dog timer is also provided in case the bootstrapping fails so that another working processor can be assigned the task of booting up the computer.

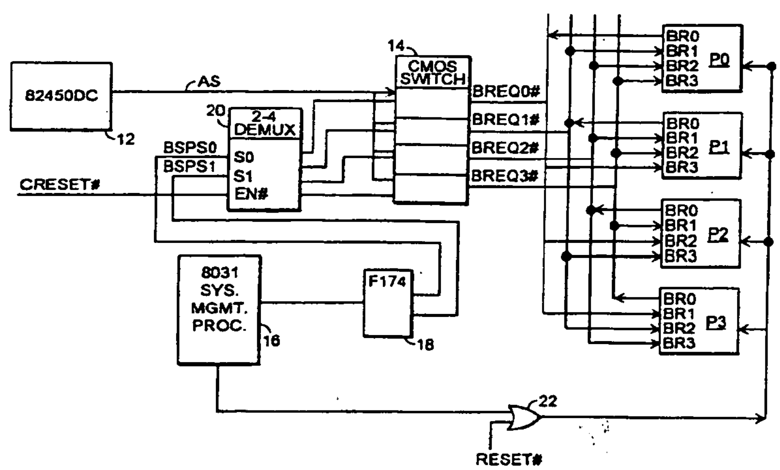

FIG. 2

EP 0 818 736 A2

Description

BACKGROUND OF THE INVENTION

Multiprocessor computers implement varying levels of symmetry. Master-slave processor computer systems are very asymmetric, whereas in computers designed with higher levels of symmetry, each of the working processors are capable of performing the same functions. In symmetric computers, the working processors share buses, address the same memory and basic input/output system (BIOS) resources, and receive the same array of interrupts.

Symmetric multiprocessor computers also tend to have a non-hierarchal bus arbitration process. The Intel® Pentium-Pro® Processor Bus arbitration protocol, for example, supports fair arbitration between a maximum of four working processors in the typical multiprocessor configuration. A unique identification, i.e., agent ID of 0, 1, 2, or 3, is assigned to each working processor as part of the power-on configuration by a central agent that is also responsible for bus arbitration control. Bus arbitration between the working processors is then made using a round-robin algorithm, which arranges the working processors in a circular order of priority: 0, 1, 2, 3, 0, 1, 2, etc. The working processor with the highest priority becomes the bus owner on every arbitration event, and the working processor that owns the bus is allowed to request its use. Input/output and possibly memory subsystems may, however, have higher priority, defeating the immediate use of the bus by the working processor.

The symmetry in some multiprocessor computers tends to break down when considering reset protocol in which the computer is bootstrapped after being powered-on or rebooted after an operating system failure. Pentium-Pro® multiprocessor systems fall into this class. The same working processor (called the boot strap processor) performs the bootstrapping operations every time the computer is powered-on. The bootstrapping can generally not be assigned among the other working processors even though they have access to the same computer system resources and would thus be capable of performing the role of boot strap processor.

SUMMARY OF THE INVENTION

The asymmetry in the reset process has undesirable artifacts. It is not entirely uncommon for at least one of the working processors in a multiprocessor computer to fail. If the failed working processor is also the assigned boot strap processor then the computer as a whole is inoperable. This risk is unacceptable for expensive multiprocessor computers performing time critical functions.

In one sense, however, the computer's failure should be avoidable. It is common for these computers to have a system management processor that monitors environmental conditions and shut-down events. From this information, the system management processor many times can predict the failure of the working processor, to which bootstrapping is assigned. Because of the asymmetry and rigidity of the computers in prior art designs, however, bootstrap processor reassignment cannot occur.

The invention, in its broad form, resides in a boot strap assignment system in a symmetric multiprocessor computer, and a method therefor, as recited in claims 1 and 9 respectively. A preferred embodiment described hereinafter features a boot strap assignment method and system for a multiprocessor computer. The computer is symmetric at least to the extent that each of the working processors of the computer may perform the role of the boot strap processor. This role is assigned among the working processors by a central agent as part of the power-on configuration. An example of the central agent is a bus arbitration controller. The assignment system involves a system management processor which monitors the operation of the multiprocessor computer and controls a switching circuit that selectively transmits the boot strap assignment from the central agent to the working processors.

In specific embodiments, the switching circuit comprises a plurality of switches that intercept the boot strap assignment signal transmitted from the central agent to bus request lines passing between the working processors. A demultiplexor is used to decode boot strap selection signals from the system management processor. One of the switches is then selectively closed to pass the boot strap assignment to only one of the bus request lines.

As part of the embodiment, a watch dog timer function may be further included. This timer is then reset upon finishing bootstrapping the multiprocessor computer. Should the timer time-out after measuring a predetermined amount of time, however, the system management processor assumes a bootstrap failure. Accordingly, the system management processor assigns a new working processor as the boot strap processor utilizing the inventive switch circuit.

In general, according to another aspect, the invention may also be characterized as a method of assigning the bootstrapping function among multiple processors in a multiprocessor computer. The method includes intercepting the boot strap assignment signal from the central agent while determining which one of the multiple processors should perform the bootstrapping. The signal is then transmitted to the working processors to effect the selection.

BRIEF DESCRIPTION OF THE DRAWINGS

A more detailed understanding of the invention may be had from the following description of an exemplary embodiment, given by way of example and to be under-

stood in conjunction with the accompanying drawing, wherein:

- FIG. 1 is a block diagram showing the prior art method for distributing arbitration phase bus request signals among the four processors of a multiprocessor computer;
- FIG. 2 is a block diagram showing a circuit for controlling the boot strap processor assignment process; and
- FIG. 3 is a process diagram illustrating boot strap processor assignment and possible reassignment in the event of bootstrapping failure according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 shows the prior art bus arbitration aspects of a Pentium-Pro® multiprocessor computer having four working processors P0, P1, P2, P3.

Each of the working processors includes four bus request pins BR0, BR1, BR2, BR3 that are associated with bus arbitration. The BR0 bus request pin is an input/output pin on each of the working processors and is used to signal arbitration requests for the bus during normal operation. The BR1, BR2, BR3 bus request pin of each working processor are input only pins, being used to monitor the arbitration requests of the respective other working processors.

Bus request lines BREQ0#, BREQ1#, BREQ2#, BREQ3# are distributed among the four working processors P0, P1, P2, P3 in a rotating fashion. The BR0 pin of the first processor P0, for example is connected to the bus request line BREQ0#; the BR0 pin of the second processor P1 is connected to the BREQ1# bus request line, and so on. In a similar vein, the BR1 pin of the first processor P0 is connected to the BREQ1# bus request line; the BR1 pin of the second processor P1 is connected to the BREQ2# bus request line. The remaining lines and pins are connected in this fashion.

During bus arbitration, the bus owner among the working processors P0, P1, P2, P3 will arbitrate the bus via its BR0 pin. The bus request is passed to the BR1, BR2, BR3 pins of the other working processors via the bus request lines. In this way, each working processor P0, P1, P2, P3 can send its bus requests to the other working processors while monitoring the other processors requests.

Bus ownership, which is coordinated via the bus request lines, is rotated among the working processors based upon the agent ID. Each working processor maintains its agent ID, assigned as part of the power-on configuration, and an ID that indicates the current bus owner.

During power-on configuration of the computer, every working processor P0-P3 is assigned its unique agent ID. A central agent 12, the bus arbitration controller, is responsible for asserting the BREQ0# bus signal during reset. On the active to inactive transition of the computer reset signal RESET#, each of the working processors P0-P3 samples its BR1, BR2, BR3 pins to determine its corresponding agent ID for arbitration. In the illustrated example, processor P0 would have an agent ID of 0 since none of its BR1, BR2, and BR3 pins are active, and processor P3 would have an agent ID of 3 since its BR1 pin is active. The agent ID cannot change without the re-assertion of the RESET#.

The working processor that is assigned the agent ID of 0 recognizes that it is the boot strap processor. After the RESET# signal goes inactive, this assigned boot strap processor will automatically begin bootstrapping the computer and in time will bring the other working processors on-line. Since processor P0 must always receive the agent ID of 0, it is always the boot strap processor.

FIG. 2 shows a modification to the bus arbitration system that enables the control of the above-described boot strap processor assignment process.

The assignment signal on line AS from the bus arbitration system controller 12 is intercepted by switch circuit 14. The circuit 14 preferably uses CMOS switches to lower electrical loading and reduce delay time. This allows the timing requirements of the original specifications to be met.

The switch circuit 14 selectively transmits the assignment signal only to one of the bus request lines BREQ0#, BREQ1#, BREQ2#, BREQ3# during the assertion of the RESET# signal. This process has the effect of controlling the assignment of the agent IDs among the working processors P0, P1, P2, and P3, thus selectively dictating the boot strap processor.

The switch circuit 14 is controlled by a 8031 system management processor 16 through a latch 18 and demultiplexor 20. The system processor 16 performs a watch dog timer function in which it initiates its own reset of the working processors P0, P1, P2, P3 via OR gate 22, which also receives the system reset signal RESET#. The system management processor 16 is also connected to transducers, not shown, distributed around the computer to monitor environmental conditions such as temperature and line voltage. Additionally, it has non-volatile persistent storage and can access the main logic boards of the computer to interrogate the previously saved shut-down events.

FIG. 3 is a process diagram illustrating the boot strap assignment of the present invention utilizing the switch circuit 14 of Fig. 2. Based upon the available information such as previous, shut-down events and environmental conditions of the computer, the system management processor 16 determines which of the working processors P0, P1, P2, and P3 is functioning properly and is thus qualified to perform the bootstrapping operation in step 110.

The selection signal from the system management processor 16 is stored in the latch 18. The two-bit output

BSPS0, BSPS1 of the latch 18 is provided to the S0 and S1 ports respectively of a 2:4 demultiplexer 20. Based upon this selection data, the demultiplexer 20 closes only one of the normally open CMOS switches 14 in response to receiving CRESET# signal at its enable port EN# in step 112. It therefore selects one of the bus request lines to receive the assignment signal AS from the bus arbitration controller 12. CRESET# is a delayed version of the system reset signal RESET# so that it is still valid on the active to inactive transition of the system reset signal RESET #, which causes the processors to sample their respective BR1-BR3 pins.

The selection process of the switches 14 determines the agent IDs, step 114, for the working processors P0-P3 since the assignment signal as is provided on the selected one of BREQ0# through BREQ3# lines. Based upon the established agent ID assignment protocol, the working processor whose BR0 pin is active and whose BR1, BR2 and BR3 pins remain inactive assumes the agent ID of 0, and takes the role as the boot strap processor. The working processor whose BR1 pin is active assumes the agent ID of 3; the working processor whose BR2 pin is active assumes the agent ID of 2; and the working processor whose BR3 pin is active assumes the agent ID of 1.

Shortly after the system reset signal RESET# is inactive, CRESET# is also de-asserted. When this occurs, the demultiplexor 20 is no longer enabled causing the CMOS switches 14 to open. Thus, the bus request lines BREQ0# through BREQ3# become isolated and will not be influenced during normal operation by residual electrical loading from the added circuitry of the CMOS switches 14.

The assigned bootstrap processor accesses the boot address space in step 116. It initializes all functions on the system, finishes diagnostics, and wakes up the remaining working processors one at a time to ensure that they are properly functioning. The boot strap processor also updates a table that is accessible to the operating system indicating which of working processors are operating properly.

In parallel with the boot strapping of the computer, the system management processor 16 also initializes a watchdog timer function to begin counting-down from a predetermined time period in step 117. In the preferred embodiment, the time period is five seconds.

The final role of the assigned bootstrap processor is to signal the system management processor to reset the watchdog timer function. The resetting of the watchdog timer at the end of the bootstrapping indicates to the system management processor 16 that the bootstrapping operation was successful and that operation is proceeding normally.

It may occur, however, that the bootstrapping operation fails because of the failure of the bootstrapping processor. In this case, the boot strap processor will never reach the end of the bootstrapping operation and signal the system management processor 16. The watchdog timer will time-out indicating unsuccessful bootstrapping in step 120.

When the watchdog timer times-out, the system management processor 16 initiates a second reset operation and assigns the bootstrapping function to another processor by asserting its own reset to the working processor via the OR gate 22 with appropriate control of CRESET#, step 122. An indication of this newly assigned processor is stored in latch 18, and steps 111 through 117 are repeated as illustrated by loop 200 in Fig. 3. Further, system management processor 16 identifies and labels the working processor that failed in the bootstrapping operation as inoperative. The newly assigned boot strap processor subsequently updates the table to indicate the failed processor to the operating system.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the scope of the invention.

Claims

1. A boot strap assignment system in a symmetric multiprocessor computer including a plurality of working processors each of which may serve as a boot strap processor and a central agent which assigns one of the working processors to be the boot strap processor as part of power-on configuration, the boot strap assignment system comprising:

   a switching circuit which selectively transmits a boot strap assignment generated by the central agent to the working processors such that one of the working processors is assigned to serve as the boot strap processor.

2. A boot strap assignment system as described in Claim 1, further comprising a system management processor which monitors the operation of the multiprocessor computer and controls the switching circuit to determine which of the working processors receives the boot strap assignment.

3. A boot strap assignment system as described in Claim 2, wherein the switching circuit comprises a plurality of switches that intercept a boot strap assignment signal from the central agent to bus request lines passing between the working processors and selectively passes the boot strap assignment signal to only one of the bus request lines.

4. A boot strap assignment system as described in Claim 3, wherein the switching circuit further comprises a demultiplexor that decodes a boot strap processor selection signal from the system man-

agement processor and selectively closes one of the switches to pass the boot strap assignment signal to the selected one of the bus request lines.

5. A boot strap assignment system as described in Claim 3, wherein each of the working processors includes four bus request pins, one of the bus request pins being used to signal the other working processors over the bus request lines of a request to use a bus that is common to the working processors and the other bus request pins being used to detect bus requests from the other working processors.

6. A boot strap assignment system as described in Claim 1, further comprising a watch dog timer which the boot Strap processor resets upon finishing bootstrapping the multiprocessor computer.

7. A boot strap assignment system as described in Claim 6, wherein the system management processor resets the multiprocessor computer and selects a new working processor to serve as boot strap processor in response to the watch dog timer timing-out after the previous boot strap processor fails to boot strap the multiprocessor computer.

8. A multiprocessor computer comprising:

   a plurality of working processors connected to each other via rotated bus request lines, each of the working processors being capable of bootstrapping the computer;
   a bus arbitration controller which arbitrates access to a bus for the working processors and which generates a boot strap assignment signal as part of power-on configuration that assigns responsibility for bootstrapping the computer among the working processors; and
   a switching circuit, coupled between the bus arbitration controller and the bus request lines of the working processors, the switching system selectively transmitting the boot strap assignment signal generated by the bus arbitration controller to the working processors via the bus request lines.

9. A method of assigning bootstrapping among multiple working processors in a multiprocessor computer in which a central agent generates a boot strap assignment signal, the method comprising

   intercepting the boot strap assignment signal from the central agent;
   determining which of the multiple working processors should perform the bootstrapping; and
   transmitting the boot strap assignment signal among the working processors so that the determined one of the working processors is signaled to perform the bootstrapping.

10. A method as described in Claim 9, further comprising transmitting the boot strap assignment signal to one of a plurality of bus request lines that are rotated between the working processors.

11. A method as described in Claim 9, wherein transmitting the boot strap assignment signal to the working processors comprises:

   receiving the boot strap assignment signal at a plurality of switches that connect to the separate bus request lines that pass between the working processors; and
   selectively closing one of the switches to pass the boot strap assignment signal to only one of the bus request lines.

12. A method as described in Claim 9, further comprising determining which of the multiple working processors should perform bootstrapping in response to environmental conditions for the computer and previous shut-down events.

13. A method as described in Claim 9, further comprising generating a boot strap assignment signal in a bus arbitration controller.

14. A method as described in Claim 9, further comprising:

   activating a watch dog timer in response to reset; and
   resetting the watch dog timer upon finishing bootstrapping the multiprocessor computer.

15. A method as described in Claim 14, further comprising assigning a new processor as the boot strap processor in response to the watch dog timer timing-out after the previous boot strap processor failed to boot strap the multiprocessor computer.

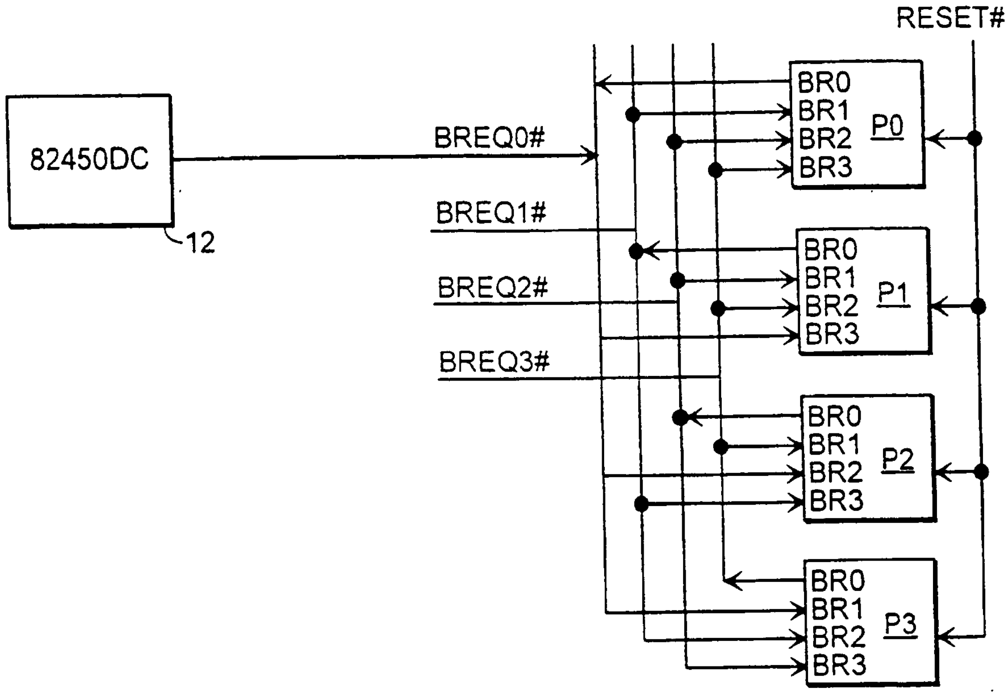

FIG. 1

(PRIOR ART)

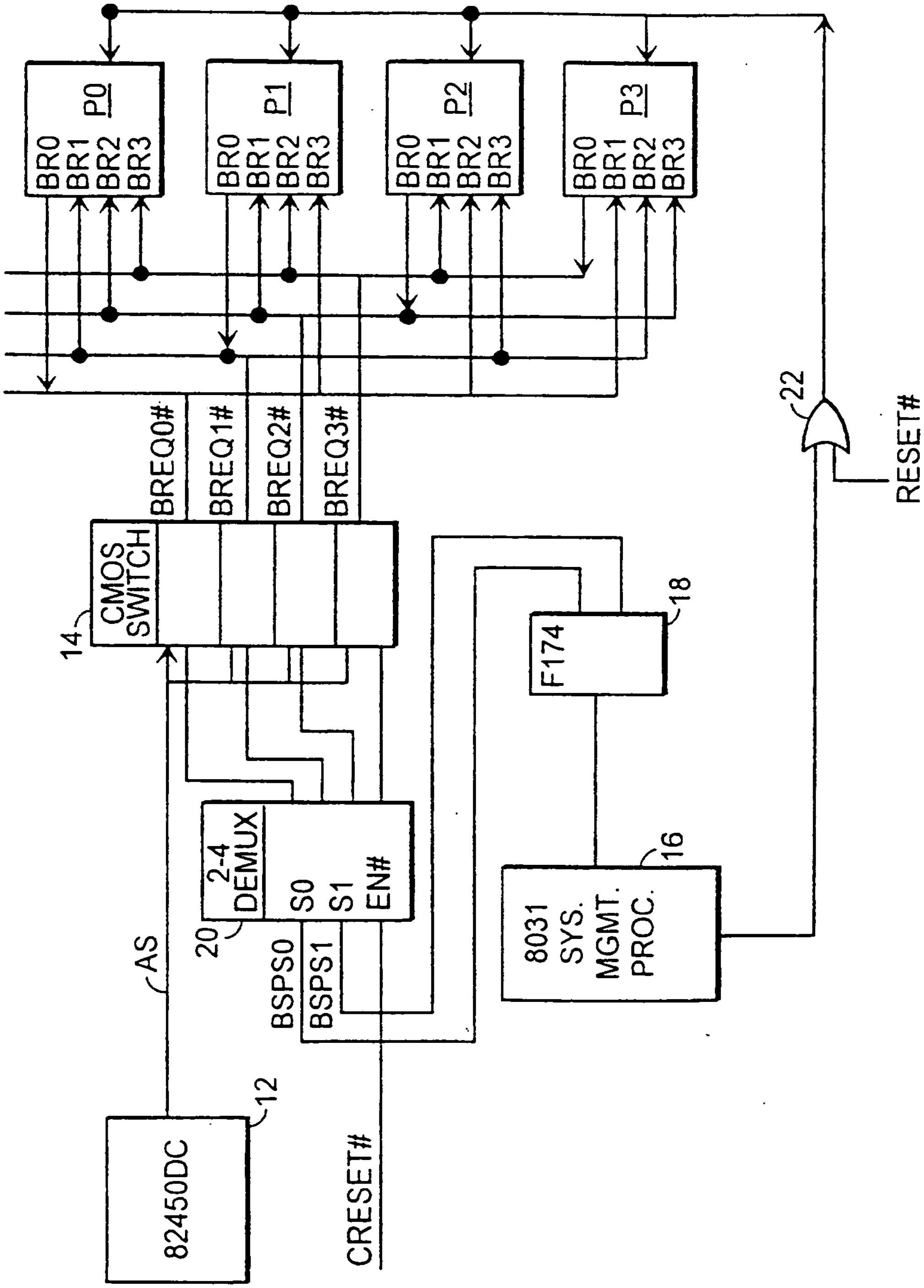
P0
BR0
BR1
BR2
BR3
P1
BR0
BR1
BR2
BR3
P2
BR0
BR1
BR2
BR3
P3
BR0
BR1
BR2
BR3
BREQ0#
BREQ1#
BREQ2#
BREQ3#
CMOS SWITCH
14
22
RESET#
F174
18
2-4 DEMUX
S0
S1
EN#
20
AS
BSPS0
BSPS1
8031 SYS. MGMT. PROC.
16
82450DC
12
CRESET#

FIG. 2

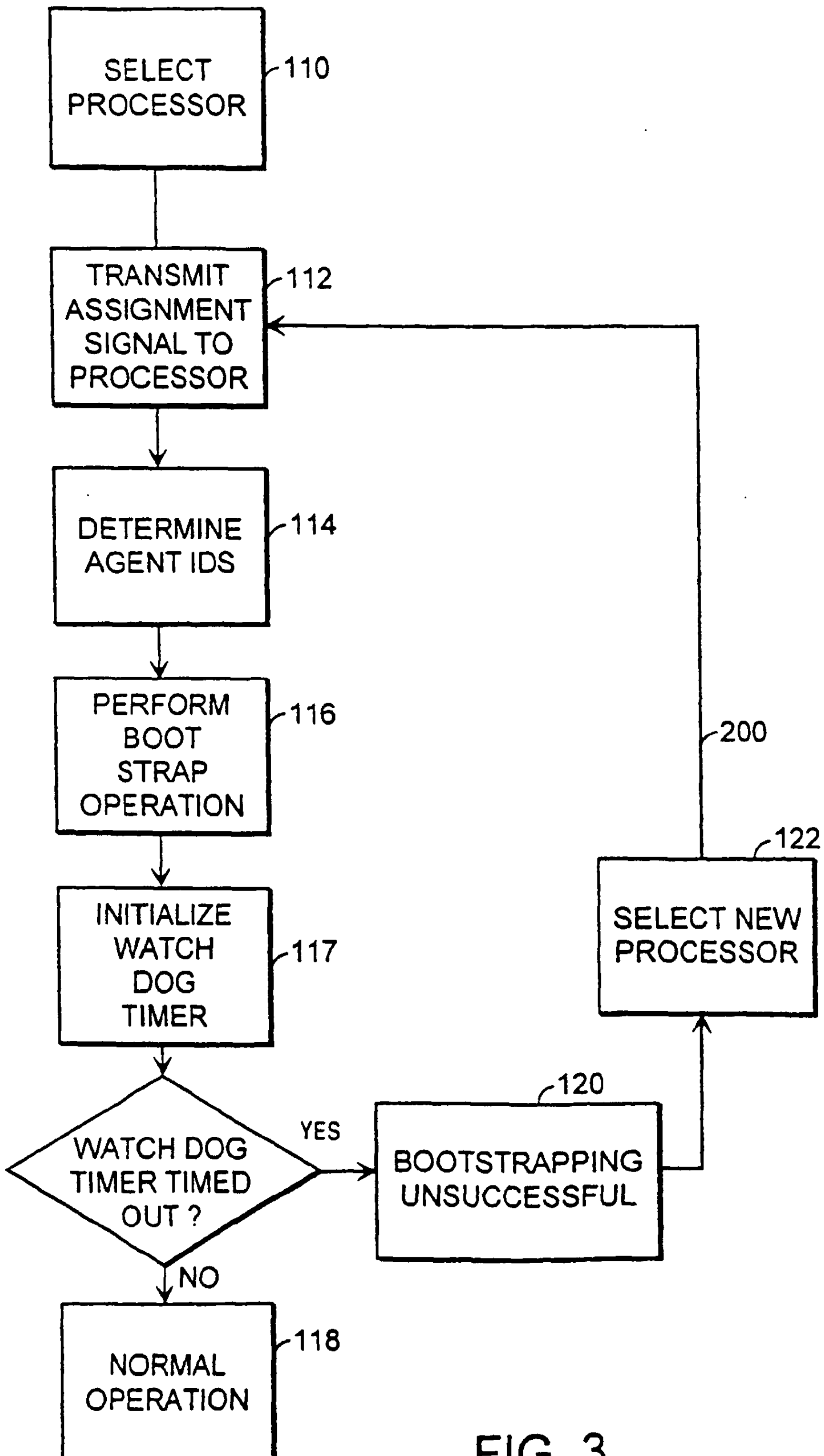
SELECT PROCESSOR
110
TRANSMIT ASSIGNMENT SIGNAL TO PROCESSOR
112
DETERMINE AGENT IDS
114
PERFORM BOOT STRAP OPERATION
116
INITIALIZE WATCH DOG TIMER
117
WATCH DOG TIMER TIMED OUT ?
YES
NO
BOOTSTRAPPING UNSUCCESSFUL
120
SELECT NEW PROCESSOR
122
200
NORMAL OPERATION
118

FIG. 3